

项目三 创建和维护表



项目导入

通过项目二的学习，相信大家都已经创建好 student 数据库了，相关的数据（学生、课程、教师等）放在哪儿呢？比较方便的方法是通过 SQL Server 2000 进行数据的存储。在 SQL Server 2000 中，数据的存储是通过二维表来实现的。



项目要求

- 掌握数据库表的创建方法。
- 学会用 T-SQL 语句创建表、修改表和删除表。
- 熟悉几种常用的约束的创建方法。
- 学会对表数据进行插入、修改和删除。



项目准备

一、关系（二维表）

由于 SQL Server 2000 是关系型数据库管理系统，所以数据和数据之间的关系是以一张关系（二维表）存储。

基本概念：关系。

关系就是数据与数据之间的联系，它必须符合以下条件：

- (1) 每一列的数据类型都必须相同。
- (2) 没有完全相同的行。
- (3) 列与列之间的字段名不能重复。
- (4) 没有组合列，即在一列中没有子列，或说不能把这一列分割成两列或更多列。

基本概念：基本表（Table）。

基本表就是一个关系，由行（记录）和列（字段）组成的二维列表，是 SQL Server 的主要数据库对象。

SQL Server 中有两种基本表：系统表（以 sys 作为前缀的表）和用户表。

二、SQL Server 2000 中数据类型

基本概念：数据类型的三个术语。

- (1) 精度：即为数值型数据整数部分和小数部分的和。如 1234.12 的精度为 6。

(2) 小数位数：即为数值型数据小数部分位数。

(3) 长度：存储数据所占用的字节数。

SQL Server 中提供了丰富的系统数据类型，常用的系统数据类型如表 3-1 所示。

表 3-1 SQL Server 2000 常用的数据类型

数据类型	符号标识
整型	int, smallint, tinyint
浮点型	Real, float, decimal, numeric
字符型	Char, varchar, nchar, nvarchar
日期和时间型	datetime, smalldatetime
文本和图形型	Text, ntext, image
货币型	money, smallmoney
位数据类型	Bit
二进制数据类型	binary, varbinary
特殊数据类型	timestamp, uniqueidentifier
新增数据类型	bigint, sql_variant, table

1. 整型

int (integer) 类型占用 4 个字节；smallint 类型占用 2 个字节；tinyint 类型占 1 个字节。

2. 浮点型

浮点型用于存储十进制小数，采用只入不舍的方式。

(1) real 类型：占用 4 个字节，最大 7 位精确位数。

(2) float 类型：可以精确到第 15 位小数，默认占用 8 个字节的存储空间。

表示为 float (n)：n 取 1~7，用 4 个字节存储；n 取 8~15，用 8 个字节存储。

(3) decimal 和 numeric：可以提供小数所需要的实际存储空间，用 2~17 个字节存储。

表示为 decimal(p,s)，其中 p 为总宽度，s 为小数宽度。

3. 字符型

字符型用来存储各种字母、数字符号和特殊符号。

字符型标识符：' ' (英文单引号) 或 " " (英文双引号)。

(1) char 类型：占用 1 个字节。表示为：char (n) (n 取 1~8000，默认为 1)。

(2) varchar 类型：可以存储长达 8000 个字符的可变长度字符串。表示为：varchar (n) (n 根据输入数据的实际长度变化)。

(3) nchar 类型：用于存储固定长度的 Unicode (统一字符编码标准) 数据。表示为：nchar (n) (n 取 1~4000) (每个 Unicode 字符占用 2 个字节)。

(4) nvarchar 类型：用于存储可变长度的 Unicode (统一字符编码标准) 数据。表示为：nvarchar(n) (n 取 1~4000) (每个 Unicode 字符占用 2 个字节)。

4. 日期和时间型

(1) `datetime` 类型：用于存储日期和时间的结合体，占用 8 个字节。

存储范围：1753 年 1 月 1 日零时起至 9999 年 12 月 31 日止。

存储精度：三百分之一秒，即 3.33 毫秒。

默认格式：`MM DD YYYY hh:mm A.M./P.M.`

(2) `smalldatetime` 类型：用于存储 1900 年 1 月 1 日至 2079 年 6 月 6 日内的日期，占用 4 个字节。

5. 文本和图形型

(1) `text` 类型：用于存储字符超过 8000 个的文本数据。

(2) `ntext` 类型：用于存储字符超过 4000 个且为可变长度的 Unicode 文本数据；理论容量为 $2^{30}-1$ (1,073,741,823) 个字节。

(3) `Image` 类型：用于存储超过 8000 个且为可变长度的二进制数据，适合存储照片、目录图片或者图画；理论容量为 $2^{31}-1$ (2,147,483,647) 个字节。

6. 货币型

(1) `money` 类型：用于存储货币值，占用 8 个字节，精度为货币单位的万分之一。

(2) `smallmoney` 类型：其用于存储货币值，占用 4 个字节。

注意：货币型数据必须在有效位置前面加一个货币单位符号（如\$或其他货币单位的记号）。

7. 位数据类型

`Bit` 类型：相当于其他语言中的逻辑型数据，只存储 0 和 1 或 NULL。

8. 二进制数据类型

(1) `binary` 类型：用于存储固定长度的 n (1~8000) 个字节的二进制数据；字段长度为 $n+4$ 个字节；常用于存储图像等数据。

(2) `varbinary` 类型：用于存储可变长度的 n (1~8000) 个字节的二进制数据；字段长度为实际所输入数据的长度加上 4 字节。

注意：在输入二进制常量时，需在该常量前面加一个前缀 `0x`。

9. 特殊数据类型

(1) `timestamp` 类型：自动记录时间的时间戳数据类型，主要用于在数据表中记录其数据的修改时间，此时间在整个数据库范围内是唯一的。

(2) `uniqueidentifier` 类型：唯一标识符数据类型；系统为此产生全局唯一标识符代码。

10. 新增数据类型

(1) `bigint` 类型：占用 8 个字节。

(2) `sql_variant` 类型：用于存储除文本、图形数据和 `timestamp` 类型数据外的其他任何合法的 SQL Server 数据。

(3) `table` 类型：用于存储对表或者视图处理后的结果集。

项目 3.1 创建表结构

“表”是数据库的一个最重要的对象。可以通过企业管理器或查询分析器创建表结构。

任务 1 使用企业管理器创建表结构



提出任务

在 student 数据库下，使用“企业管理器”创建 xs（学生）表对象，如表 3-2 所示。

表 3-2 （学生）表结构

字段名	数据类型	长度	允许为空	字段含义
xh	char	8	not null	学号
xm	char	8		姓名
xb	char	2		性别
csrq	datetime			出生日期
zy	char	20		专业



分析任务

在“企业管理器”中“展开 student 数据库，选择“表”右击，在弹出的快捷菜单中选择“新建表”即可创建表结构。



完成任务

(1) 进入“企业管理器”，展开 student 数据库，选择“表”，在右边窗口中右击，选择“新建表”命令，弹出“表设计器”窗口，“表设计器”窗口如图 3-1 所示。

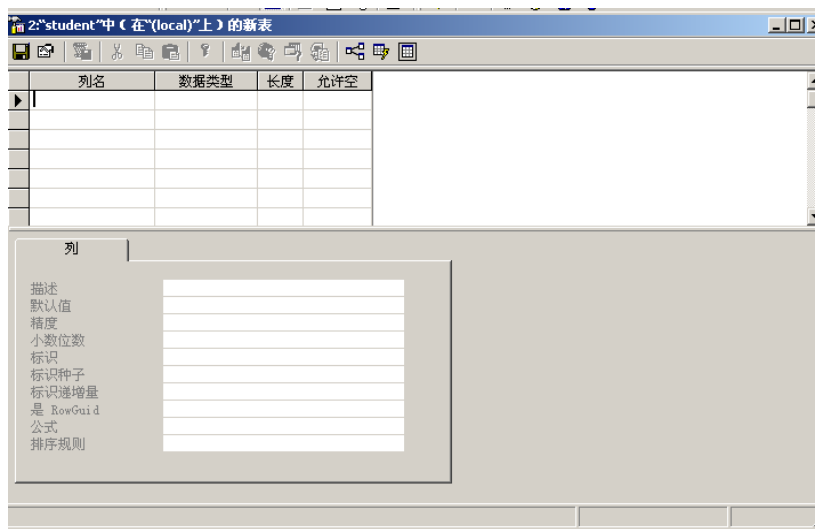


图 3-1 student 数据库中新表“表设计器”窗口

(2) 在图 3-1 所示的“表设计器”的上部分表格中输入列名、数据类型、长度和允许空字段属性，其中前三个是必须输入的，字段属性输入后的界面如图 3-2 所示。

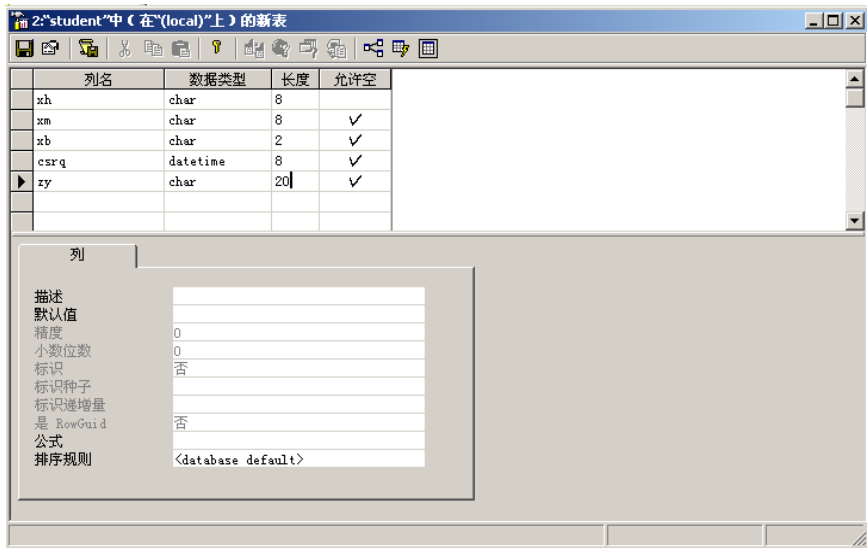


图 3-2 student 数据库中 xs 表“表设计器”窗口

注意：

对于 decimal 和 numeric 数据类型的字段，可以指定该字段的精度和小数位数。

对于整型、decimal 和 numeric 数据类型的字段，可以设置标识列，使其具有自动编号功能。

任务 2 使用查询分析器创建表结构

基本语法：CREATE-TABLE 语句的语法格式。

CREATE TABLE 表名

(**字段 1 数据类型** [not null] [identity(seed,increment)][,···n])

基本参数如下：

- 常见符号[]：代表可省略。
- 加粗字体代表最基本语法格式。
- “···n”：可以有 n 个字段。



提出任务

使用“查询分析器”创建项目 3.1 的 xs 表结构。



分析任务

在 T-SQL 中，创建表结构是通过 CREATE TABLE 语句创建的，直接在“查询分析器”中输入 T-SQL 代码即可。



完成任务

(1) 进入“查询分析器”，选择 student , CREATE TABLE 语句代码为：

```
CREATE TABLE xs
```

```
(xh char(8) not null, xm char(8), xb char(2), csrq datetime, zy char(20))
```

(2) 单击工具栏上的 按钮分析代码，如果出现“命令已成功完成”，代表代码正确，单击 ，执行此代码。



项目练习

练习 3.1 若表中字段类型为 char，长度为 20，当输入字符串“计算机及应用”时，此字段占用_____字节存储空间。若字段类型为 varchar，将占用_____字节存储空间。

练习 3.2 在 student 数据库中创建 kc（课程），cj（成绩）表对象。

cj（成绩）表结构如表 3-3 所示。

表 3-3 cj（成绩）表结构

字段名	数据类型	长度	小数位	允许空	字段含义
xh	char	8		not null	学号
kcbh	char	3			课程编号
cj	decimal	4（精度）	1	not null	成绩

kc（课程）表结构如表 3-4 所示。

表 3-4 kc（课程）表结构

字段名	数据类型	长度	小数位	允许空	字段含义
kcbh	char	3		not null	课程编号
kcm	char	20			课程名
xf	int				学分

思考：“学生”表已创建好，如何保证学生记录的正确性和完整性？很显然，我们应该对表数据进行一些限制来确保数据的质量。在 SQL Server 2000 中是通过约束来实现的。

项目 3.2 使用企业管理器创建完整性约束

一、完整性约束的类型

基本概念：完整性约束的类型。

(1) 实体完整性。实体完整性要求数据库表中的每一条记录都是唯一的，即表中没有重复的记录。因此每个实体都必须拥有一个主键或唯一性约束。

(2) 域完整性：域完整性要求存入字段中的数据值必须符合特定的条件来保证数据的有效性。

(3) 引用完整性：引用完整性涉及两个或两个以上表的数据的一致性维护。当添加、删除或修改数据库表中的记录时，可以借助引用完整性来保证相关联的表之间的数据一致性。

它们之间的关系如图 3-3 所示。

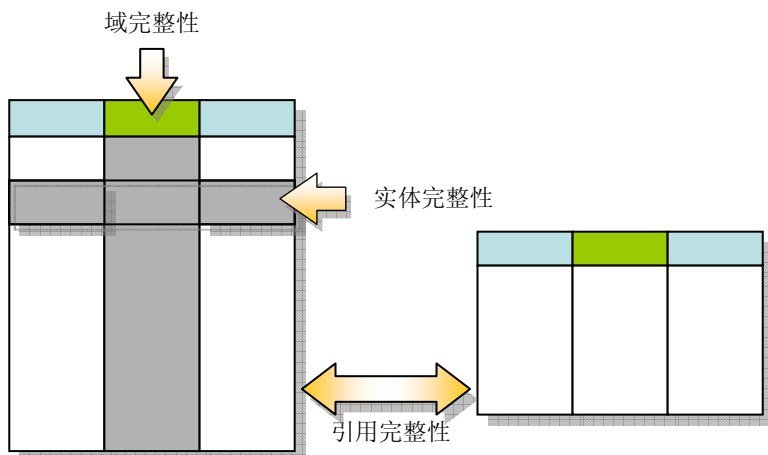


图 3-3 完整性约束之间的关系

二、SQL Server 的约束

基本概念：SQL Server 的约束。

(1) 主键 (Primary Key) 约束：就是基本表中的主码，用来唯一标识每行。每个表只能有一个主键，且主键不能为空，不能重复。

(2) 外键 (Foreign Key) 约束：用于维护同一数据库中两表之间的一致性关系，使用户在一张表中做增删查改操作时要注意与其关联的另一张表。

(3) 默认 (Default) 约束：用于指定一个字段的默认值。当用户未指定字段数据时，系统自动指派的数据值。

(4) 检查 (Check) 约束：使用逻辑表达式来限制字段上可以接受的数据值。检查约束通过检查输入表中字段的数据值来维护域完整性。

(5) 唯一性 (Unique) 约束：主要用于非主键字段约束。每张表可以定义多个唯一性约束，定义了唯一性约束的字段值不可重复。

任务 1 创建主键 (Primary Key) 约束




提出任务

student 数据库中有 xs (学生)、kc (课程)、cj (成绩) 三个数据库对象，按要求创建如下主键约束：

1. 将 xs 表中的 xh (学号) 字段设置为主键。
2. 将 cj 表中的 xh (学号) 和 kcdh (课程代号) 两个字段组合起来作为主关键字。




分析任务

由题意, 打开 xs 和 cj 表结构, 先将相关字段设置成 NOT NULL, 然后单击工具栏中  按钮即可设置主键。



完成任务

1. 操作步骤

- (1) 进入“企业管理器”, 打开 xs (学生) 表的“表设计器”窗口。
- (2) 检查 xs (学生) 表中的 xh (学号) 字段是否为 NOT NULL, 若满足条件, 在 xh (学号) 的行选择器处单击。
- (3) 单击工具栏或快捷菜单中的  按钮, xs (学生) 表的主关键字设置完成, 界面如图 3-4 所示。

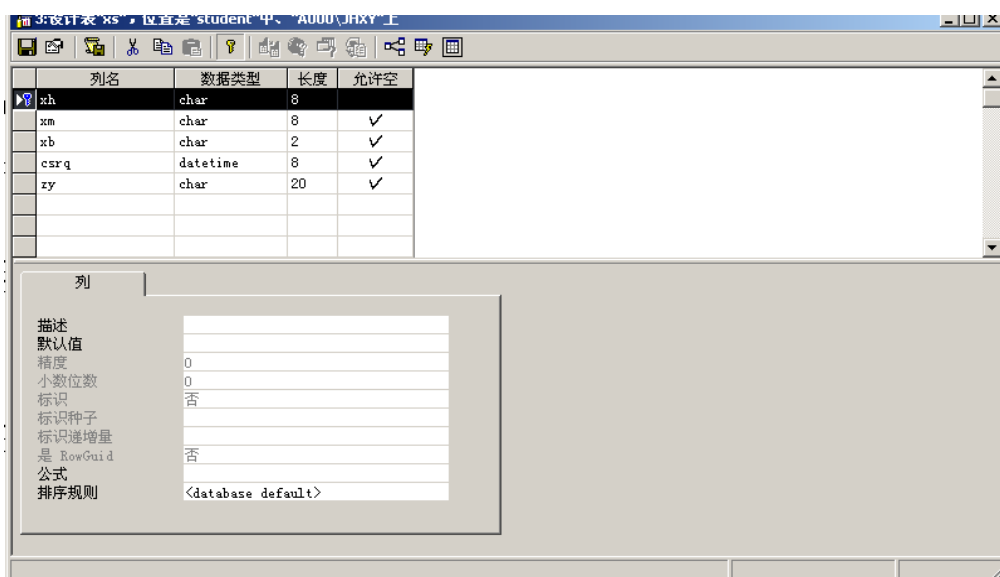



图 3-4 创建 xs 表的主键约束

2. 操作步骤

- (1) 进入“企业管理器”, 打开 cj (成绩) 表的“表设计器”窗口。
- (2) 检查 cj (成绩) 表中的 xh (学号) 和 kcdh (课程代号) 两个字段是否为 NOT NULL, 若满足条件, 按住 Ctrl 键, 在 xh 和 kcdh 的行选择器处依次单击进行多选。
- (3) 单击工具栏或快捷菜单中的  按钮, cj (成绩) 表的主关键字设置完成, 界面如图 3-5 所示。

思考: xs 表和 cj 表都有共同字段 xh (学号), 它们之间应该怎么联系?

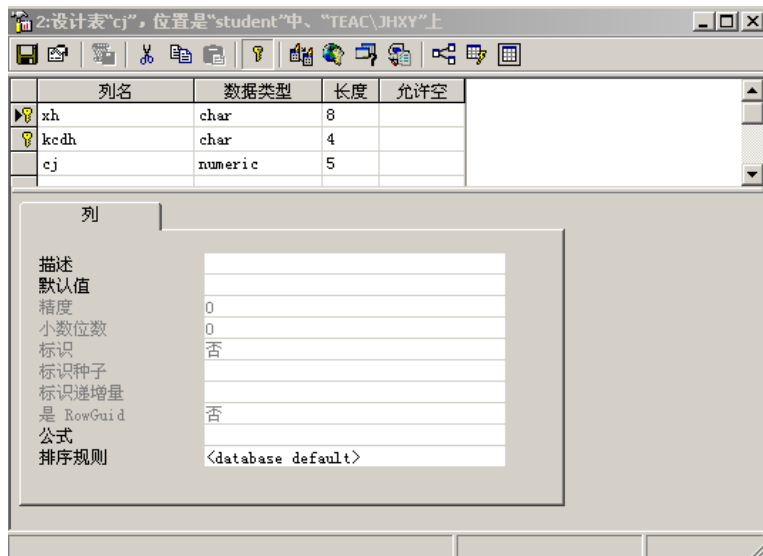


图 3-5 创建 cj 表的主键约束

任务 2 创建外键（Foreign Key）约束



提出任务

将 xs 表中的 xh（学号）字段设置为 cj 表的外键。




分析任务

由题意，以 xs 表为主表，cj 表为子表，创建 xh 字段的外键约束。




完成任务

方法一：使用“管理关系”按钮创建外关键字。

- (1) 进入“企业管理器”，打开 xs（学生）或 cj（成绩）表的“表设计器”窗口。
- (2) 选择 （管理关系）按钮，将出现“属性”对话框，选择“新建”按钮，在“主键表”中选择 xs 表的 xh 字段，在“外键表”中选择 cj 表的 xh 字段，属性界面如图 3-6 所示。

方法二：使用数据库关系图创建外关键字。

- (1) 进入“企业管理器”，展开 Student 数据库；选择  关系图 节点右击，在弹出的快捷菜单中选择“新建数据库关系图”命令，弹出如图 3-7 所示的“创建数据库关系图向导”对话框。
- (2) 单击“下一步”按钮，选择 xs（学生）和 cj（成绩）表。
- (3) 单击“完成”按钮。

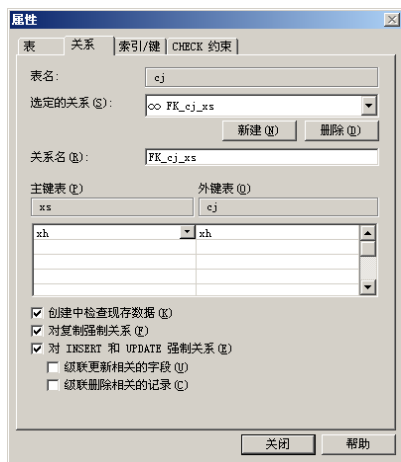


图 3-6 创建 cj 表的外键约束

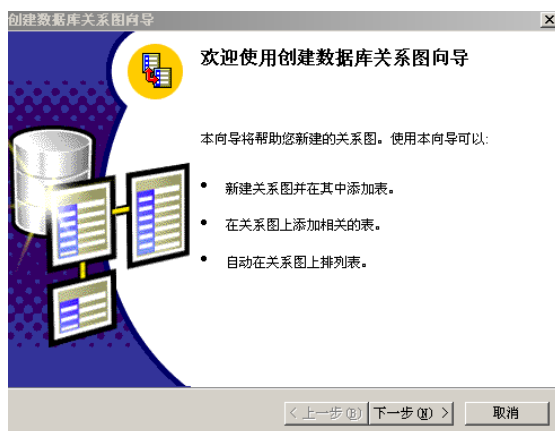


图 3-7 “创建数据库关系图向导”对话框

(4) 选择 xs 表的 xh (学号) 字段, 按住鼠标左键不放, 拖到 cj 表 xh 字段, 出现如图 3-8 所示的“创建关系”对话框。

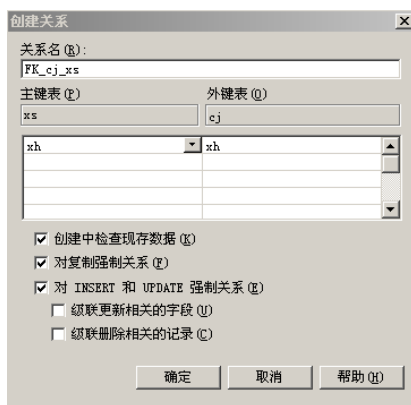


图 3-8 “创建关系”对话框

(5) 建立好的外键约束图如图 3-9 所示。

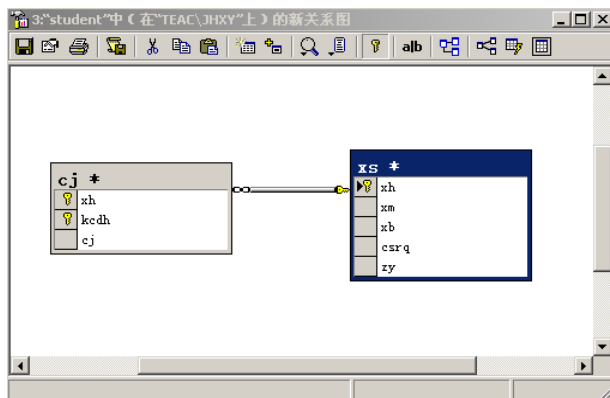


图 3-9 外键约束图

技巧：对于字段组合作为主关键字的情况（如成绩表），在实际设计过程中通常添加一个字段作为标识列。

任务3 创建默认值（Default）约束



提出任务

1. 将 xs 表中的 xb（性别）字段默认值设置为“男”。
2. 为 xs 表中的 zy（专业）设置默认值约束：“计算机应用技术”。



分析任务

由题意，打开 xs 表结构，分别选择 xb 字段和 zy 字段，在“默认值”框里输入内容即可。



完成任务

1. 操作步骤

(1) 进入“企业管理器”，打开 xs（学生）表的“表设计器”窗口。

(2) 选中 xs（学生）表中的 xb（性别）字段，在“默认值”框里输入“男”，如图 3-10 所示。

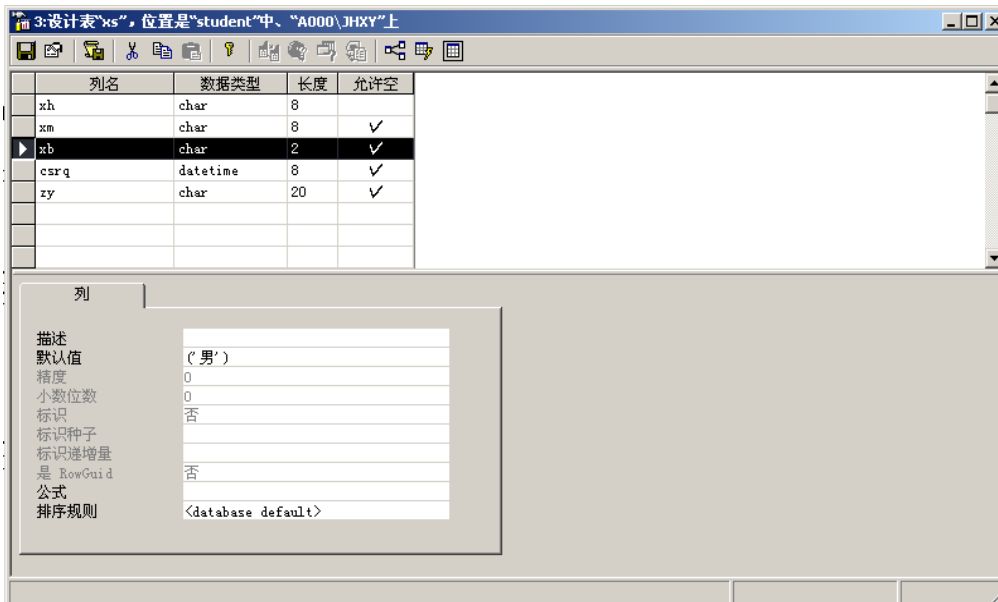


图 3-10 创建 xb 字段的默认值约束

(3) 单击“保存”按钮。

2. 操作步骤

- (1) 进入“企业管理器”，打开 xs（学生）表的“表设计器”窗口。
- (2) 选中 zy（专业）列，在“默认值”框里输入“计算机应用技术”，如图 3-11 所示。

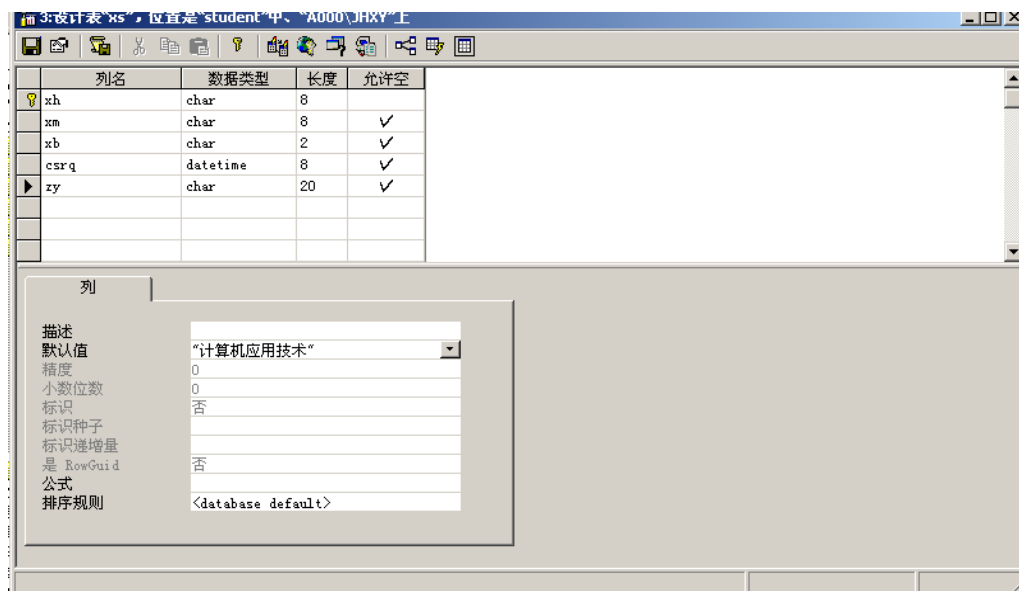


图 3-11 创建 zy 字段的默认值约束

- (3) 单击“保存”按钮。

任务 4 创建检查（Check）约束



提出任务

为 cj 表中的 cj（成绩）字段设置检查约束：成绩在 0~100 之间。




分析任务

由题意，打开 cj 表结构右击，在弹出的快捷菜单中选择“属性”命令，在“CHECK 约束”选项卡中进行设置。



完成任务

- (1) 进入“企业管理器”，打开 cj（成绩）表的“表设计器”窗口。
- (2) 右击，在弹出的快捷菜单中选择“属性”命令，或者单击  按钮，进入“属性”对话框，在“约束表达式”中输入“cj>0 and cj<100”，如图 3-12 所示。
- (3) 单击“关闭”按钮，“CHECK 约束”已创建好。

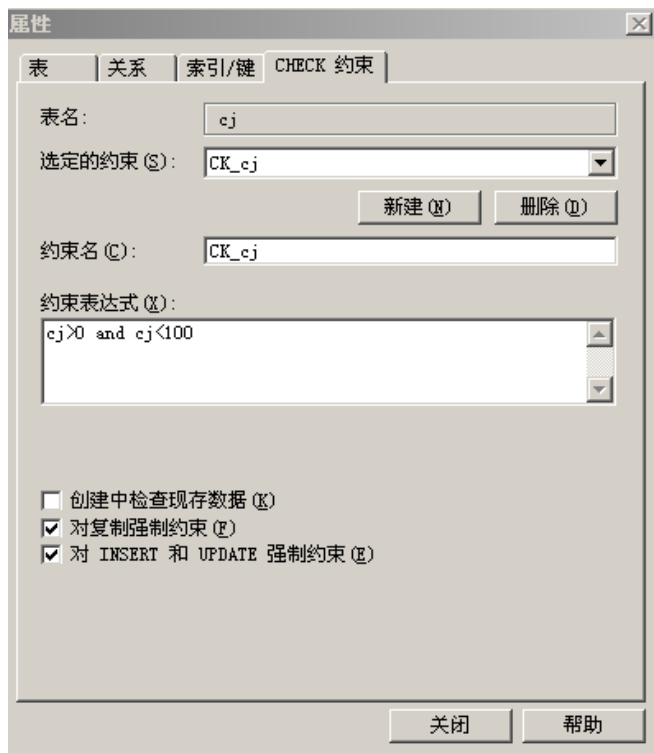


图 3-12 创建 cj 表的 CHECK 约束

任务 5 创建唯一性 (Unique) 约束



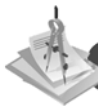
提出任务

为 xs 表中的 xm (姓名) 字段设置唯一性约束。




分析任务

由题意, 打开 xs 表结构右击, 在弹出的快捷菜单中选择“属性”命令, 在“索引/键”选项卡中进行设置。



完成任务

- (1) 进入“企业管理器”, 打开 xs (学生) 表的“表设计器”窗口。
- (2) 右击任一字段所在的行, 选择“属性”命令, 或者选择  快捷按钮, 进入“索引/键”选项卡, 单击“新建”按钮, 在“列名”框中选择 xm, 勾选“创建 UNIQUE”复选框, 选择“约束”单选按钮, 如图 3-13 所示。
- (3) 单击“关闭”按钮, “索引/键”已创建好。

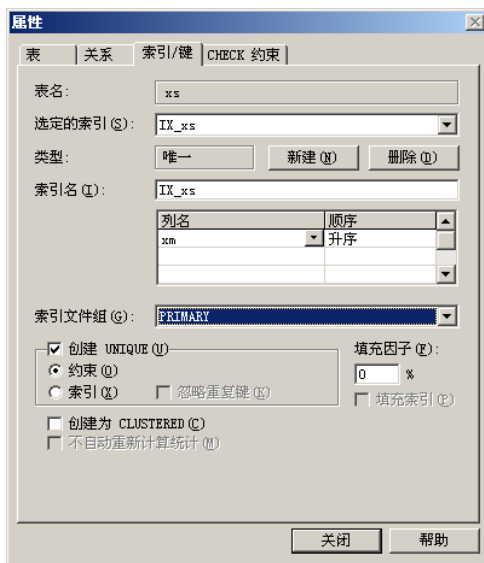


图 3-13 创建 xs 表的唯一性约束

项目 3.3 使用 T-SQL 语言创建完整性约束

任务 1 通过 CREATE-SQL 重新创建完整性约束

基本语法： 带有约束条件的 CREATE-SQL 语句的语法格式。

CREATE TABLE 表名

(字段 数据类型 列的特征 [,...n])

“列的特征”具体参数描述：

- 空值 (NULL) 约束: [NULL | NOT NULL]
- 默认值 (DEFAULT) 约束:

[[CONSTRAINT 约束名] DEFAULT 默认值]

- 主键 (PRIMARY KEY) 约束:

[[CONSTRAINT 约束名]

PRIMARY KEY [CLUSTERED | NONCLUSTERED] [(字段名,...n)]

- 外键 (FOREIGN KEY) 约束:

[[CONSTRAINT 约束名] FOREIGN KEY [(字段名,...n)]

REFERENCING 引用表(引用字段[,...n])]

- 唯一性 (UNIQUE) 约束:

[[CONSTRAINT 约束名] UNIQUE

[CLUSTERED | NONCLUSTERED] [(字段名,...n)]

- 检查 (CHECK) 约束:

[[CONSTRAINT 约束名] CHECK (约束条件)]



提出任务

根据例项目 3.2 中各表的约束条件,使用带有约束条件的 CREATE-SQL 语句重新创建 xs、kc、cj 表结构。



分析任务

由题意,删除 xs、kc、cj 表,在“查询分析器”中输入 CREATE-SQL 语句重新创建。



完成任务

(1) 创建 xs 表的 T-SQL 语言如下:

```
CREATE TABLE xs
(
    xh char (8) NOT NULL PRIMARY KEY,
    xm char(8) CONSTRAINT IX_xs UNIQUE NONCLUSTERED,
    xb char (2) CONSTRAINT DF_xs_xb DEFAULT ('男'),
    csrq datetime,
    zy char (20) CONSTRAINT DF_xs_zy DEFAULT ('计算机应用技术'),
)
```

(2) 创建 kc 表的 T-SQL 语言如下:

```
CREATE TABLE kc
(
    kcbh char (3) NOT NULL PRIMARY KEY,
    kcm char (20) ,
    xf int,
)
```

(3) 创建 cj 表的 T-SQL 语言如下:

```
CREATE TABLE cj
(
    xh char (8) NOT NULL
    CONSTRAINT FK_cj_xs
    FOREIGN KEY REFERENCES xs(xh),
    kcdh char (3) NOT NULL ,
    cj decimal(4, 1) NOT NULL
    CONSTRAINT CK_cj CHECK (cj>= 0 and cj <= 100),
    CONSTRAINT PK_cj PRIMARY KEY (xh, kcdh)
)
```

说明: CREATE TABLE-SQL 语句中加粗部分为完整性约束语句。

思考: 如果 xs、kc、cj 表中有记录就不能先删除表,再重新创建表结构,那么如何修改表结构呢?

任务 2 通过 ALTER-SQL 修改表结构

基本语法：ALTER-TABLE 语句的语法格式。

(1) 修改字段属性：

```
ALTER TABLE 表名 ALTER COLUMN 字段名 类型 [字段特征]
```

(2) 增加字段：

```
ALTER TABLE 表名 ADD 字段名 类型 [字段特征]
```

(3) 删除字段或约束：

```
ALTER TABLE 表名 DROP COLUMN | CONSTRAINT 字段名 | 约束名
```

(4) 添加约束：

```
ALTER TABLE 表名 ADD CONSTRAINT 字段名 [字段特征]
```



提出任务

根据项目 3.2 中各表的约束条件，使用 ALTER-SQL 语句修改 xs、kc、cj 表结构。



分析任务

由题意，在“查询分析器”中输入 ALTER-SQL 语句修改表结构。



完成任务

(1) 项目 3.3 中任务 1 对应的 ALTER-SQL 代码：

```
ALTER TABLE xs  
ADD CONSTRAINT PK_xs PRIMARY KEY CLUSTERED (xh)  
ALTER TABLE cj  
ADD CONSTRAINT PK_cj PRIMARY KEY CLUSTERED (xh,kcdh)
```

(2) 项目 3.2 中任务 2 对应的 ALTER-SQL 代码：

```
ALTER TABLE cj  
ADD CONSTRAINT FK_cj_xs FOREIGN KEY (xh) REFERENCES xs (xh)
```

(3) 项目 3.2 中任务 3 对应的 ALTER-SQL 代码：

```
ALTER TABLE xs  
ADD CONSTRAINT DF_xs_xb DEFAULT ('男') FOR xb  
ALTER TABLE xs  
CONSTRAINT DF_xs_zy DEFAULT ('计算机应用技术') FOR zy
```

(4) 项目 3.2 中任务 4 对应的 ALTER-SQL 代码：

```
ALTER TABLE cj  
ADD CONSTRAINT CK_cj CHECK (cj>= 0 and cj <= 100)
```

(5) 项目 3.2 中任务 5 对应的 ALTER-SQL 代码：

```
ALTER TABLE xs  
CONSTRAINT IX_xs UNIQUE NONCLUSTERED (xm)
```


项目练习

练习 3.2

- (1) 将 kc 表中的 kcdh (课程代号) 字段设置为主键。
- (2) 将 kc 表中的 kcdh (课程代号) 字段设置为 cj 表的外键。
- (3) 将 kc 表中的 xf (性别) 字段默认值设置为 3。
- (4) 为 kc 表中的 kcm (课程名) 字段设置唯一性约束。

解题思路: 可以通过企业管理器或查询分析器完成的 kc (课程) 表的完整性约束。

思考: 表结构已经创建好了, 如何对表记录进行添加、修改、删除? 在 SQL Server 2000 中, 通过什么操作对数据进行管理?

项目 3.4 管理表中数据

对表进行快速添加、修改、删除数据操作, 同样可以通过企业管理器和查询分析器进行设置。

任务 1 使用企业管理器管理表数据

提出任务

对 student 数据库中的 xs 表:

1. 在 xs 表中输入一些数据, 参考数据如图 3-14 所示。

xh	xm	xb	csrq	zy
071101	卞伟	男	1988-1-1	计算机应用技术
071105	陈维扬	男	1988-3-5	计算机应用技术
071107	伏依违	男	1988-6-8	计算机应用技术
071109	侯超	男	1989-4-7	计算机应用技术
071111	黄一伦	男	1989-5-2	计算机应用技术
071143	耿敏	女	1987-6-9	计算机应用技术
071144	龚香	女	1989-8-6	计算机应用技术
071146	刘华春	女	1988-7-8	计算机应用技术
071147	苗青	女	1988-9-5	计算机应用技术
071150	许迪	女	1988-12-3	计算机应用技术
085201	经鑫	男	1989-8-9	应用日语
085207	陈立问	男	1988-5-19	应用日语
085232	杨茜	女	1988-3-6	应用日语
085233	辛小燕	女	1989-5-6	应用日语

图 3-14 xs 表中输入的数据

2. 在 xs 表中, 将“计算机应用技术”专业改为“计算机及应用”专业。
3. 在 xs 表中, 删除所有“应用日语”专业的学生。

分析任务

由题意, 进入“企业管理器”, 选中“表”右击, 在弹出的快捷菜单中选择“打开表”→“返回所有行”命令进行设置。



完成任务

1. 操作步骤

(1) 进入“企业管理器”，选中 xs（学生）表右击，在弹出的快捷菜单中选择“打开表”→“返回所有行”命令，进入 xs 表的表数据窗口。

(2) 在数据窗口中输入记录。

2. 操作步骤

(1) 进入“企业管理器”，选中 xs（学生）表右击，在弹出的快捷菜单中选择“打开表”→“返回所有行”命令，进入 xs 表的表数据窗口。

(2) 若记录很少，定位到“计算机应用技术”记录，把它改成“计算机及应用”专业。

(3) 若记录很多，打开“SQL 窗格”，输入 UPDATE-SQL 语句。

3. 操作步骤

(1) 进入“企业管理器”，选中 xs（学生）表右击，在弹出的快捷菜单中选择“打开表”→“返回所有行”命令，进入 xs 表的表数据窗口。

(2) 若记录很少，定位到“应用日语”记录右击，在弹出的快捷菜单中选择“删除”命令，此时出现“确认”对话框，单击“是”按钮删除所选记录。

(3) 若记录很多，打开“SQL 窗格”，输入 DELETE-SQL 语句。

任务 2 使用查询分析器管理表数据

基本语法：

(1) 使用 INSERT-SQL 语句插入新记录：

INSERT [INTO] 表名[(字段列表)] VALUES (值列表)

(2) 使用 UPDATE-SQL 语句修改记录：

UPDATE 表名 SET 字段名=表达式 | DEFAULT | NULL [...n]
[WHERE 条件表达式]

(3) 使用 DELETE-SQL 语句删除记录：

DELETE [FROM] 表名 [WHERE 条件表达式]



提出任务

将上面的操作采用“查询分析器”实现。



分析任务

上一任务可以分别用 INSERT-SQL、UPDATE-SQL、DELETE-SQL 实现。



完成任务

进入“查询分析器”，代码为：

(1) INSERT INTO xs (xh, xm, xb, csrq, zy)
VALUES ('071101', '卞伟', '男', '1988-1-1', '计算机应用技术')

(2) UPDATE xs SET zy='计算机及应用'
WHERE zy='计算机应用技术'

(3) DELETE FROM xs
WHERE zy='应用日语'

思考：使用 INSERT-SQL 语句只能添加一条记录，使用 UPDATE-SQL 语句只能对一个表中数据行修改，如何使管理表数据的功能更加强大一些？

【扩展】

(1) 扩展 INSERT-SQL 语句：

INSERT [INTO] 表名[(字段列表)] SELECT 语句

(2) 扩展 UPDATE-SQL 语句：

UPDATE 表名 SET 字段名=表达式 | DEFAULT | NULL [,...n]
FORM 基表[WHERE 条件表达式]

(3) 扩展 DELETE-SQL 语句：

DELETE [FROM] 表名 FORM 基表 [WHERE 条件表达式]

说明：采用扩展的 INSERT-SQL、UPDATE-SQL、DELETE-SQL 语句可以进行批量插入、修改和删除记录。可以通过 FROM 子句获取相关数据。相关知识请大家参阅相关资料。



项目练习

练习 3.3 什么是数据完整性？简述关系数据库的几种完整性，并各举一个例子。

练习 3.4

(1) 在 kc 和 cj 表中添加多条记录，内容如图 3-15 所示。

xh	kcdh	cj
071101	001	82
071105	002	69
071107	003	86
071109	004	87
071111	005	89
071143	003	76
071144	002	63
071146	001	82
071147	005	91
071150	002	83
085201	003	66
085207	002	63
085232	002	25
085233	004	69
*		

kcbh	kcm	xf
001	计算机应用基础	3
002	大学英语	4
003	高等数学	4
004	C语言	3
005	大学物理	3
*		

图 3-15 在 kc 和 cj 表中添加记录

(2) 将 kc 表中的 kcdh（课程代号）为 001 的课程增加 1 学分。

(3) 删除 xs 表中 xh（学号）为 20030101 的学生信息。



项目总结

表是一种重要的数据库对象。它是存储、管理数据以及建立关系数据库的基本单元。表中行称为记录，列称为字段。表中每个字段有一个唯一的字段名，每个字段也只有一个唯一的字段类型。

SQL Server 中常用的数据类型有：整型、浮点型、字符型、日期和时间型、文本和图形型、货币型、位数据类型、二进制数据类型、特殊数据类型和新增数据类型。

在 SQL Server 中可以用企业管理器或 T-SQL 语句管理表。企业管理器方式操作简单，命令方式更为灵活。修改表的结构时，若表是一个空表，可以任意修改它的结构；如果表中已经存储了记录，建议不要轻易修改表结构，特别不要改变数据类型，以免产生错误。

在数据库中保证数据的完整性是很重要的。所谓完整性，就是指存储在数据库中数据的一致性和正确性。实施数据库完整性的目的是确保数据的质量。SQL Server 提供了一些在字段上强制数据完整性的机制，它们是：空值（NULL）约束、默认值（DEFAULT）约束、主键（PRIMARY KEY）约束、外键（FOREIGN KEY）约束、唯一性（UNIQUE）约束和检查（CHECK）约束。

表结构建立好后，可以向表中添加、修改和删除数据。可以在企业管理器中“打开表”进行操作，也可使用 INSERT、UPDATE、DELETE 语句进行修改。



项目实训

一、实训目的和要求

- 熟练掌握在企业管理器中创建表的方法。
- 掌握使用 T-SQL 语句创建表。
- 掌握创建常见的几种约束。
- 学会通过企业管理器和 T-SQL 语句管理表数据。

二、实训内容

(1) 通过企业管理器和 T-SQL 语句创建 Bookmanage 数据库中 3 个表对象。3 个表的结构如表 3-5 至表 3-7 所示。

表 3-5 Ts (图书) 表

列名	数据类型	长度	允许空	说明
Tsbh	Char	6		图书编号
Tsmc	Varchar	40	√	图书名称
Tslb	Char	10	√	图书类别
Zz	Varchar	20	√	作者
Cbrq	Datetime	8	√	出版日期
Cbsmc	Varchar	50	√	出版社名称

续表

列名	数据类型	长度	允许空	说明
Sh	Varchar	20	√	书号
Dj	Monery	8	√	定价
Zt	Bit	1	√	状态
rgsj	datetime	8	√	入馆时间

表 3-6 Dz (读者) 表

列名	数据类型	长度	允许空	说明
Jszh	Char	5		借书证号
Xm	Varchar	10	√	姓名
Xb	Char	2	√	性别
Telephon	Varchar	13	√	电话
Bj	Varchar	20	√	班级
Jscs	tiyint	1	√	借书册数
Bzrq	datetime	8	√	办证日期
Xh	Char	10	√	学号

表 3-7 Jy (借阅) 表

列名	数据类型	长度	允许空	说明
Jyh	bigint	8		借阅号
Jszh	Char	5		借书证号
Tsbh	Char	6		图书编号
Jsrq	datetime	8		借书日期
Hsrq	datetime	8	√	还书日期
Fj	decimal	5	√	罚金

- (2) 将 ts (图书表) 的 tsbh (图书编号) 字段设置为主键。
 - (3) 将 dz (读者表) 的 jszh (借书证号) 字段设置为主键。
 - (4) 将 jy (借阅表) 的 tsbh (图书编号) 和 jszh (借书证号) 两个字段的组合作为主键。
 - (5) 将 ts (图书表) 中的 tsbh (图书编号) 设置为 jy (借阅表) 的外键。
 - (6) 将 dz (读者表) 中的 jszh (借书证号) 设置为 jy (借阅表) 的外键。
 - (7) 为 ts (图书表) 中的 dj (定价) 字段设置检查约束: 定价必须大于 0。
 - (8) 将 ts (图书表) 中的 tslb (图书类别) 设置默认值约束: 不知属哪类。
- 创建好上述 3 个表后, 输入一些记录, 验证以上约束。